

### **In The Claims**

Applicant submits below a complete listing of the current claims, with insertions, if any, indicated by underlining and deletions, if any, indicated by strikeouts and/or double bracketing.

#### **Listing of the Claims**

1. (Currently Amended) A method of debugging a target system using a host system connected thereto, the target system comprising a digital signal processor having associated memory comprising plural addressable locations, said target system further having a reserved storage location designated as a vector address, said memory further storing a plurality of application programs, each application program having respective associated exception handler code, the method comprising:

dynamically loading a stack to a reserved region of said memory; and

causing the vector address of said target system to point to said stack, whereby all said application programs use ~~the~~ said stack for a particular exception.

2. (Currently Amended) The method of claim 1 further comprising the steps of:  
dynamically loading a library to said target system from said host system whereby said dynamically loaded library has an entry point at one of said plural addressable locations, wherein said library includes at least one routine needed for running at least one of said application programs; and

storing information indicative of the address of said one location at a reserved location in said stack.

3. (Currently Amended) The method of claim 2 further comprising the step of:  
using said host system to start ~~of~~ one of said application programs, whereby a running application identifies the need for said at least one routine;  
reading said vector address;  
using the contents of the vector address to access said stack;  
reading said reserved stack location to derive the entry point of said library to said one application program;

calling said at least one routine from said library.

4. (Currently Amended) The method of claim 3 wherein said at least one routine comprises a communication routine enabling said host system and said target system to communicate.

5. (Currently Amended) The method of claim 3 wherein said at least one routine comprises a routine enabling a hardware bug to be worked round.

6. (Currently Amended) The method of claim 3 wherein said step of calling comprises supplying a first item of data indicative of the at least one routine and a second item of data for the operation to be performed by said at least one routine.

7. (Currently Amended) The method of claim 6 wherein said at least one routine returns an item of data to said one application program.

8. (Original) The method of claim 6 wherein each said data item comprises a machine word.

9. (Currently Amended) A device for debugging a target system, the device comprising a host system connected thereto, the target system comprising a digital signal processor having associated memory comprising a plurality of addressable locations, said target system further having a reserved storage location designated as a vector address, said memory further storing plural application programs, each application program having respective associated exception handler code, the device further comprising:

stack dynamic loading circuitry in said host system for dynamically loading a stack to a reserved region of said memory, whereby said loading circuitry comprises an indication of the location in said memory of said stack; and

vector writing circuitry receiving said indication, and writing to said vector address of said target system the address of said stack, whereby all said application programs use the said stack for a particular exception.

10. (Previously Presented) The device of claim 9 further comprising:  
a computer file in said host, said file comprising a library having a routine needed by at least one of said application programs;  
library dynamic loading circuitry for dynamically loading said library to said target from said host whereby said dynamically loaded library has an entry point at one of said plural addressable locations; and  
stack writing circuitry for storing information indicative of the address of said one location at a reserved location in said stack.
11. (Currently Amended) The device of claim 10 further comprising:  
control circuitry in said host for starting one of said application programs, whereby a running application identifies the need for said routine;  
vector reading circuitry in said target for reading the content of said vector address;  
addressing circuitry for using the contents of the vector address to access said stack;  
stack reading circuitry for reading said reserved stack location to derive the entry point of said library to said one application program;  
calling circuitry for calling said routine from said library.
12. (Original) The device of claim 11 wherein said calling circuitry is operable to supply a first item of data indicative of the routine and a second item of data for the operation to be performed by said routine.
13. (Original) The device of claim 12 wherein each said data item comprises a machine word.
14. (New) A method of debugging a target system using a host system connected thereto, the target system comprising a digital signal processor having associated memory comprising a plurality of addressable locations, the target system further having a reserved storage location designated as a vector, the memory further storing a plurality of application

programs, each application program having respective associated exception handler code, the method comprising:

- dynamically loading a library from the host system to the target system, the library including a communication routine enabling the host system and the target system to communicate;

- dynamically loading a stack to a reserved region of the memory; and

- causing the vector of the target system to point to the stack, whereby all the application programs use the stack for a particular exception.

15. (New) The method of claim 14, wherein the dynamically loaded library has an entry point at one of the plurality of addressable locations, wherein the library includes at least one routine needed for running at least one of the application programs, the at least one routine including the communication routine, the method further comprising:

- storing information indicative of the address of the one addressable location at a reserved location in the stack.

16. (New) The method of claim 15, further comprising:  
using the host system to start one of the application programs, whereby a running application identifies a need for the at least one routine;

- reading the vector;

- using the contents of the vector to access the stack;

- reading the reserved stack location to derive the entry point of the library to the one application program;

- calling the at least one routine from the library.

17. (New) The method of claim 16, wherein the at least one routine comprises a routine enabling a hardware bug to be worked round.

18. (New) The method of claim 16, wherein the step of calling comprises supplying a first item of data indicative of the at least one routine and a second item of data for the operation to be performed by the at least one routine.

19. (New) The method of claim 18, wherein the at least one routine returns an item of data to the one application program.

20. (New) The method of claim 18, wherein each of the data items comprises a machine word.